THE LOGICAL ORGANIZATION OF THE NEW IBM SCIENTIFIC CALCULATOR

M. M. Astrahan and N. Rochester

International Business Machines Corp.

The new IBM scientific calculator installation contains a large electrostatic memory and extensive input-output facilities. These fit it especially well for handling partial differential equations, large matrix operations, and a wide variety of other problems. The most significant advances in the logical organization of this system lie in the input-output devices and the manner in which they are connected to the central part of the calculator. For this reason, the central part of the machine will be described rather briefly, and the chief emphasis will be placed on the input and output.

The electrostatic memory consists of two units, each of which can hold 1024 words. Each word consists of a 35-bit magnitude and

LOCATION	INSTRUCTION	
0310	R ADD	1710
0311	SUBTRACT	1724
0312	TRANSFER +	0311
0313	STOP	0000

Figure 1

An Illustrative Program

a sign and is a little more precise than a 10-digit decimal number. A word may alternatively be interpreted as two half-words and the addressing system allows the programmer to conveniently address either the full word or the left half-word or the right half-word. Whenever 17 bits provide adequate precision, the programmer has, at his disposal, double the number of locations for data.

The electrostatic memory is used for instructions as well as data. In the portion of memory being used for program, each half-word contains one single-address instruction.

Arithmetic is done in parallel with a fixed point, but the point may be regarded as being located at any desired place. Consequently, the programmer can conveniently manipulate integers, fractions, or mixed numbers. The multiplication speed is greater than 1000 per second.

Figure 1 shows a program to illustrate the operation of the instructions. It consists of four instructions, located at addresses 0310 through 0313. These addresses refer to successive half-word locations. The first instruction, at address 0310, tells the machine to reset the accumulator and to add into it the number from address 1710. The instruction

ADD	SUBTRACT
RESET & ADD	RESET & SUBTRACT
ADD ABSOLUTE	SUBTRACT ABSOLUTE
STORE	LOAD M-Q
STORE ADDRESS	STORE M-Q
MULTIPLY	DIVIDE
MULTIPLY ROUND	ROUND
LONG LEFT	LONG RIGHT
ACCUMULATOR LEFT	ACCUMULATOR RIGHT
TRANSFER	TRANSFER ON ZERO
TRANSFER ON PLUS	TRANS ON OVERFLOW
STOP	

Figure 2

Arithmetical Instructions

at 0311 says to subtract from the accumulator the number at 1724. The next instruction says to go back to 0311 if the sign of the accumulator is positive, but to go on if the sign is negative. The calculator will go around this two-instruction loop until the accumulator becomes negative. Then it will proceed to the instruction at 0313 which says to stop.

Figure 2 gives a list of the arithmetical instructions.

Figure 3 shows the engineering model of the calculator. Up to now only the main frame and the electrostatic memory have been discussed. These are shown in the center. The memory unit now in use holds 1024 full words and a second identical unit will be available as an option. The two units at the extreme left each hold two magnetic tape

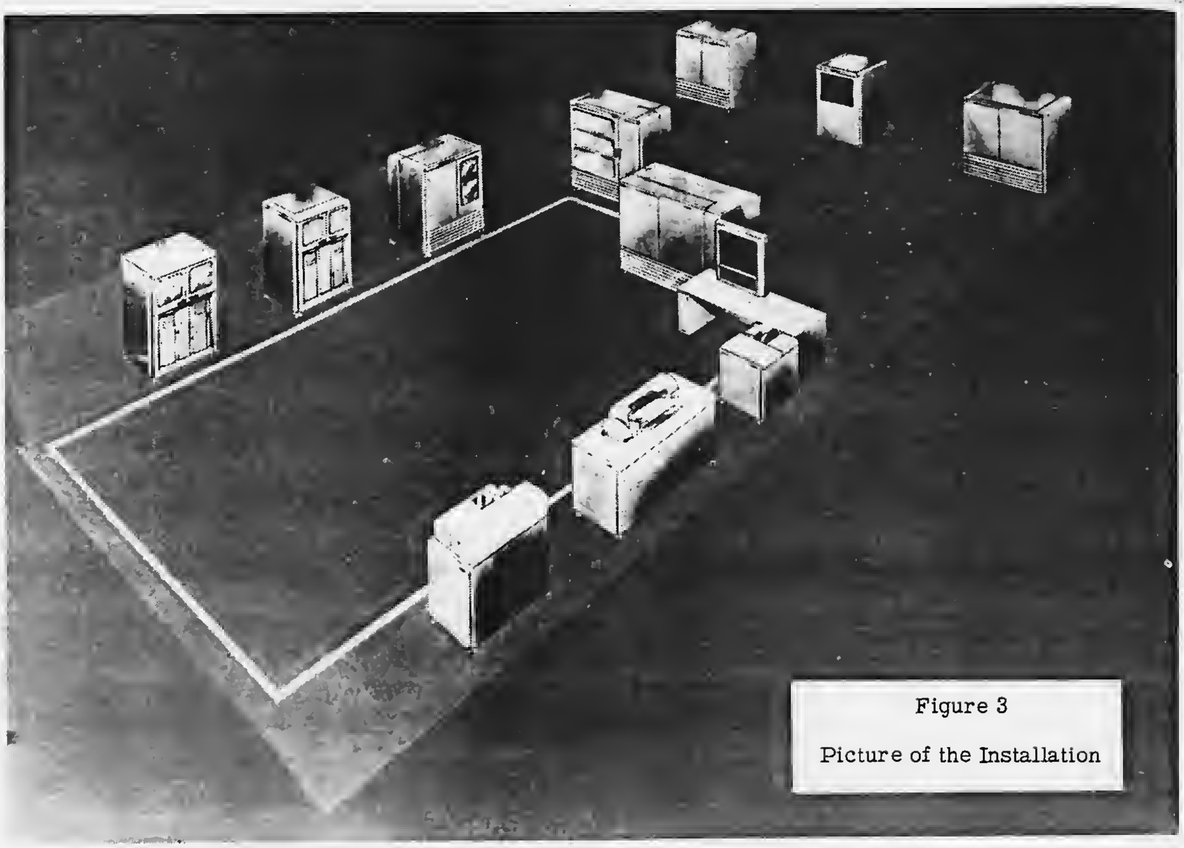

Figure 3

Picture of the Installation

drives, a total of four tapes. The next frame is the drum memory which has a capacity of 8192 full words.

On the other side of the main frame, next to the operator's panel, is the card reader. This unit can read 150 cards per minute. Next to it is the printer. The calculator can translate, print, and check seven signed ten decimal digit numbers per line at the rate of 150 lines per minute. On the extreme right is the card punch which can punch 100 cards per minute.

The calculator is enough faster than any of the input-output units or the auxiliary drum memory so that, when reading for example, it can execute several instructions after receiving one word and before receiving the next. This makes it possible to control the input-output units by means of a program on a word-by-word basis. This means that with magnetic tapes and drums one is not restricted to handling information in blocks of a fixed size, but may use blocks (called unit records) of any number of words he desires.

In order to read a unit record from the drum or an input-output unit, the program must contain a READ instruction whose address part specifies which unit is to be read. This must be followed in due course by COPY instructions. Each COPY instruction contains the address at which the next word is to be written in the memory. The COPY instruction can be used in a subprogram which automatically changes the address in the COPY instruction for each new word received. One such subprogram, called the standard Read-Write subprogram can be used to control the transfer of unit records in either direction between the electrostatic memory and any input-output unit or the drum memory.

Figure 4 shows the process of invoking the standard Read-Write subprogram in order to write a unit record on tape. In the example given in Figure 4, the unit record is to consist of 118 full words, taken in order, beginning at address 1040.

Those who prefer to do so may regard the five instructions shown as a single three-address instruction. One address specifies which input-output unit. Another specifies the location of the first word in memory, while a third specifies how many words there are in the record. When the calculator reaches this set of instructions, control will be transferred to the Read-Write subprogram. As soon as the indicated work is complete, the calculator will return from the subprogram and resume where it left off and execute the instruction at n + 6. This subprogram technique is described on Page 22 of The Preparation of Programs for an Electronic Digital Computer by M. V. Wilkes, D. J. Wheeler, and S. Gill.

Standard IBM punched cards are used in this machine. The cards can be used with the standard decimal numerical and alphabetical punching so that standard keypunches, sorters, reproducers, and so forth may be used with this new calculator. However, something new has been added. The calculator can alterna-

tively punch binary information into the cards for its own subsequent use.

Figure 5 shows an IBM card carrying 24 36-bit numbers and some control information at the right. If this card is placed in the card reader and the program reads it, the action is as follows:

1) The program calls for the card reader with the instruction READ 2048.

2) The card reader is selected and the card starts to move.

3) For a while the calculator can go on about its business, but before the first row of holes reaches the reading station, the calculator must call for the first word from the card with the instruction COPY x, where x is the address at which the first word is to be stored.

4) The card moves sidewise with the bottom (9's) edge first. When this row reaches the reading station, the information is sensed by metal brushes and the word in the first 36 columns is stored at the address which the machine has just specified.

5) A second COPY instruction must be given before this first row of holes leaves the reading station so that the calculator can get the second word which comes from columns 37 through 72.

6) Then the calculator has a breathing spell during which it can do other work before the next row is ready.

There are 24, 36-bit words on a card and each of these is equivalent to a little more than 10 decimal digits. Information is going into the calculator at a rate equivalent to 650 decimal digits per second.

Binary cards are a very convenient vehicle for programs. The card in Figure 5 carries the standard Read-Write subprogram

LOCATION	INSTRUCTION	
N		
N + 1	+ WRITE	0256
N + 2	+ R ADD	N + 2
N + 3	+ TRANS	4046
N + 4	+ STOP	1040
N + 5	+ STOP	0118
N + 6		

Figure 4

Program to Utilize Read-Write Subprogram

mentioned before. If this card is placed in the card reader and the LOAD button is depressed, the Read-Write subprogram will appear at the proper place in memory. Henceforth, one may ignore the details of reading and writing and merely invoke this subprogram as needed.

Even a long program requires only a few cards. 1000 instructions can be punched on 21 cards. Of course, programs may also be stored on tape or drums. The trouble with binary cards is that people cannot prepare them or read them as easily as they can punch or read standard IBM cards.

Figure 6 shows an IBM card with conventional numerical and alphabetical punching. In the IBM card code each digit or alphabetical character occupies one column. For example, a 2 is represented by a punch in the second row while a 6 is represented by a punch in the sixth row. Alphabetical characters use 2 punches per column. One of the easiest ways to keypunch a program is to punch one instruction per card using standard numerical punching. One advantage of this procedure is the fact that a correction can be made by merely punching a new card, and another fact is that the latest keypunches print as they punch. The program can be

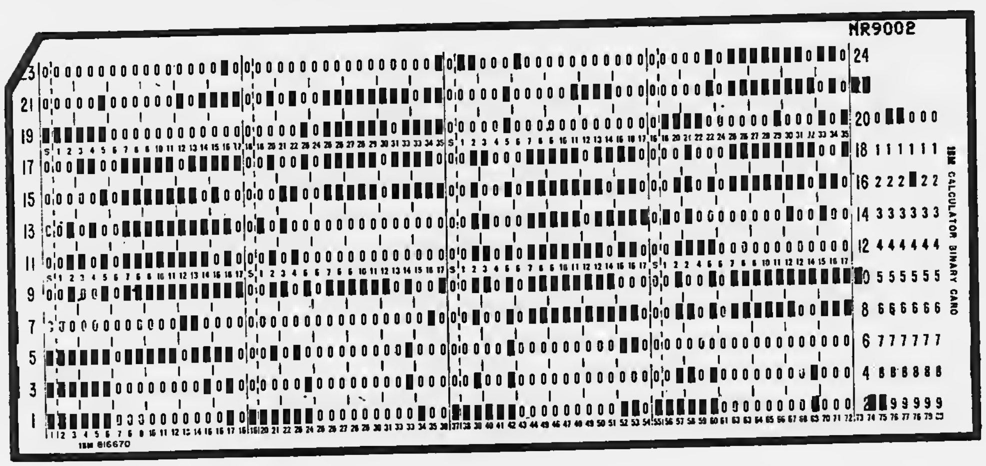

Figure 5

IBM Card with Binary Punching

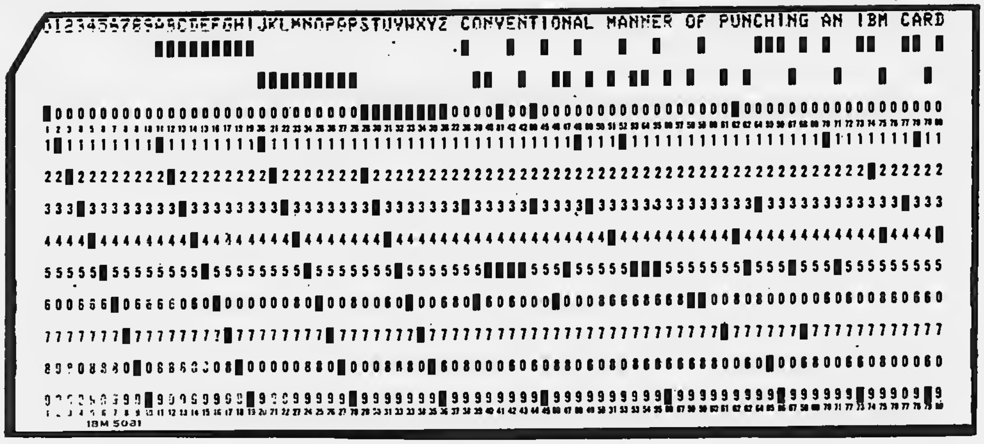

Figure 6

Conventional Manner of Punching an IBM Card

put into the calculator directly from such cards. After the program has been checked out, the calculator itself punches the program on the more compact binary cards for future use. The binary cards can then be reproduced in quantity on a standard reproducer.

In the early stages of design of this system, the plans for how to accommodate standard punching went through an evolutionary process. It seemed from the very beginning that the calculator itself should do the decimal to binary translation by means of a suitable program. An early scheme which suggested itself was to provide apparatus to convert from the decimal punching on the card to binary coded decimal and to have the calculator translate from binary coded decimal to binary. Then it became evident that the intermediate stage of binary coded decimal was neither necessary nor desirable. The calculator could as easily translate directly from decimal to binary without any specialized apparatus for the purpose.

The final scheme is to read the 9's row and regard it as a number consisting of only nines and zeros, and to translate it to binary. Then the 8's row is read and regarded as a number consisting of only eights and zeros. This is translated to binary and added to the translation of the 9's row. The process is continued until the whole card is translated.

This direct translation technique allows the calculator to read, translate, and check seven ten-digit numbers from each card at the rate of 150 cards per minute. Since full use must be made of the time between rows, the standard Read-Write subprogram is not used by the translation program. Checking of card reading is accomplished by saving and testing the card image. In other words, after reading and translating seven ten-digit numbers from a card the memory contains the seven translated numbers and the 24 words of the card image shown in Figure 5. There is time before the first row of the next card to retranslate the seven numbers back to decimal and to compare with the card image and finally to test each column of the card image to see that it contains a representation of one and only one hole per column. This test will catch single errors and any non-compensating multiple errors in card reading, translation, and retention of the numbers in memory. This checking process can be combined with the calculation of a check number which will be used in later calculation.

The printer is similar to the print unit in the IBM type 407 accounting machine. It prints 150 lines per minute and there are 120 printing positions in each line. In order to print in decimal, the program must be arranged to translate to decimal and to form a 24-word card image in memory. Then the printer is started and the successive rows of the card image are fed to the printer as needed.

The printer has a device which indicates what the position of the print wheels will be when the printing is being done. It produces impulses which come back to the calculator. The program can be arranged to retranslate these echo impulses and compare the results with the original numbers. These checks can be combined with programmed checks of the actual problem being done and with the check of card reading to provide a good overall check.

It is convenient to have the calculator print titles and other alphabetical information along with the numerical results. The calculator can therefore prepare reports in final form ready for publication as well as make the results more readable even when publication is not intended. Alphabetical printing is easily done by reading the alphabetical

information from a card and saving the 24-word card image until needed.

SUMMARY

The new IBM scientific calculator is a large-scale, high-speed, automatic calculator with an electrostatic memory capacity of 2048, 36-bit words. The parallel, binary, arithmetical unit executes more than 1000 multiplications per second. The installation includes auxiliary drum memory as well as magnetic tapes which may be used as memory or input or output. The card reader can read 150 binary or decimal cards per minute while the punch operates at 100 cards per minute. The 150 line per minute page printer can handle alphabetical as well as numerical information. Checking can be programmed to include a check on the card reading and the printing as well as the calculation.

PROCEEDINGS

OF THE

ASSOCIATION FOR COMPUTING MACHINERY

Jointly Sponsored by the Association
for Computing Machinery and the
Mellon Institute, Pittsburgh, Pa.
May 2 and 3, 1952

Published by
Richard Rimbach Associates
Pittsburgh 12, Pa.
for the
Association of Computing Machinery

1952